

### **REMARKS**

Claims 1-5, 7, 8, 10-18, and 35-54 are pending, including independent claims 1, 12, and 14. All claims are rejected as obvious over U.S. 6,895,475 to Volpe, in view of U.S. 6,128,703 to Bourekas, or over the base combination of Volpe and Bourekas, in further view of one or more additional references.

#### **Detailed Arguments regarding Independent Claims 1 and 12**

Claim 1 and claim 12 each include limitations to an integrated circuit that includes a “processor core,” “cache memory,” and additional “on-chip memory.” The claims stipulate that “on-chip memory is operable to store data for the cache memory,” that the “cache memory is filled with data from the on-chip memory for cache misses,” and that the “on-chip memory is selectably filled with data from [an] external memory independent of the cache misses under user control.”

The rejections of claims 1 and 12 state that Volpe teaches all claim limitations, except for the limitation of on-chip memory that is selectably filled with data from external memory. For this missing limitation, the examiner turns to Bourekas. On p. 5, lines 14-16, of the Office Action, the examiner states that Bourekas uses a prefetch instruction to fill secondary cache memory under user control, and that the secondary cache memory—referred to by the examiner as “second level” cache memory—serves the same purpose as the prefetch buffer in Volpe, i.e., stores data retrieved from main memory for supply to the “level one” cache memory.

These technical statements regarding the teachings of Bourekas represent clear error and all rejections fail as a matter of law because of this error. First, Bourekas teaches that the prefetch instruction fills primary cache memory, not secondary cache memory. This fact leaves the rejection argument lacking any showing of the claimed on-chip memory that is in addition to the claimed cache memory and that fills cache lines on cache misses and further is selectably filled from external memory under user control independent of cache misses.

Volpe explicitly teaches a prefetch buffer that is filled under processor control, not user control. Bourekas explicitly teaches a straight cache hierarchy, where the on-chip primary cache can be user-filled using prefetch, but Bourekas lacks any prefetch buffer, or additional on-chip memory outside of its conventional cache hierarchy. By failing to teach the claimed on-chip memory, the combination of Volpe and Bourekas fails as a matter of law to make claims 1 and 12 obvious, and thus also fails to make any of their dependent claims obvious.

All obviousness rejections based on Volpe and Bourekas further fail as a matter of law because there is no credible motivation to combine the two references. For example, the proffered motivation to combine Volpe and Bourekas first appears on p. 8 of the Office Action. There, the examiner states that it would have been obvious to combine the user-controlled prefetch teachings of Bourekas with the prefetch/cache system of Volpe, to produce an on-chip memory that:

is selectably filled with data from an external memory independent of the cache misses under user control because a prefetch instruction which specifies an ignore hit operation can prevent the microprocessor from retrieving state data from the cache even when a hit occurs, and the microprocessor does not need to flush the cache or invalidate a portion of the cache memory to insure coherency between the cache memory and the main memory....

(Emphasis added.)

The proffered motivation is not supported by the actual teachings. For example, the emphasized portions of the above motivation statement relate explicitly to how Bourekas defines and processes its user-directed prefetch operation in the context of insuring that data copied into the primary cache under user control is coherent with respect to main memory. Fig. 7 of Bourekas illustrates a "hint" field 709 appended to the prefetch instruction 704 defined by Bourekas, where the hint field tells the microprocessor whether to ignore a cache hit and access

main memory despite the access delay penalty, or to proceed with reading from the primary cache on a cache hit.

Not once in its entirety does Volpe discuss cache coherency, nor does Volpe need worry about cache coherency, at least not in the context of prefetch. That difference arises from Volpe's specific use of a prefetch buffer that is speculatively fetched into by the processor, not the user. Volpe teaches filling missed cache lines as needed, and never teaches or suggests ignoring cache hits. Volpe also teaches speculatively prefetching into its prefetch buffer under processor, not user, control. In Volpe, if a cache hit occurs, the cache data is used. There is no ability in Volpe to ignore or respond to a cache hit, based on whether user-fetched data is or is not coherent, because Volpe does not make use of user-fetched data.

The bottom line is that the proffered motivation to combine makes clear that the examiner is ignoring the actual teachings of Bourekas and Volpe, or at least seriously glossing over their differences in operation and purpose. The motivation argument abstractly pulls from Bourekas the notion of user-directed prefetching. However, as is appreciated by those skilled in the art, microprocessor prefetching necessarily depends on the supporting context and architecture. It is clear that user-directed prefetching directly into the primary cache of Bourekas does not combine with processor-directed prefetching into the prefetch buffer of Volpe.

#### Detailed Arguments regarding Independent Claim 14

Claim 14 is rejected as obvious over Volpe and Bourekas, in further combination with U.S. 5,987,590 to Wing So. The third reference, Wing So, is used only for its alleged teachings regarding a general-purpose processor and a DSP. Fundamentally, the rejection of claim 14 depends on the same erroneous Volpe-plus-Bourekas rejection arguments rebutted in the above arguments given for claims 1 and 12.

However, the rejection of claim 14 fails for the further reason that Wing So does not teach the first and second processor limitations of claim 14. Specifically, claim 14 is similar to

claims 1 and 12 in that it claims an integrated circuit including on-chip memory and cache memory, but it further claims a first processor for general-purpose processing and a second processor for data processing, within the integrated circuit. In this context, claim 14 stipulates that the on-chip memory stores data for the cache memory. In turn, the cache memory is operable to store data for the processor core of the second processor.

Wing So appears to show a PC-based computer system where a host CPU cooperates with one or more DSP devices over a PCI bus interface. All figures and text in Wing So appear to depict DSP devices ("IDSP" devices) separately from the host CPU. See, e.g., Figs. 4a, 5, and 18, which explicitly depict DSPs communicatively coupled to a CPU via PCI bus interconnects. Wing So plainly does not teach the integrated circuit arrangement of claim 14, and apparently does not teach any integrated circuit having a first processor for general purpose processing and a second processor for data processing. Instead, it apparently teaches a computer system having a CPU and one or more external, PCI-connected (card-based) DSPs, where the DSPs process instructions offloaded from the CPU when it is too busy—see, e.g., the Abstract of Wing So.

It is clear that Volpe, Bourekas, nor Wing So, taken alone, or taken in any combination, actually teaches the integrated circuit of claim 14. Indeed, a careful reading of the rejection arguments on pp. 18 and 19 make clear that the examiner realizes that the three-way combination does not teach the claimed integrated circuit. The rejection arguments also implicitly acknowledge that Wing So itself does not teach an integrated circuit including the claimed first/second processor arrangement.

Apparently, then, Wing So is used in the obviousness combination merely because it depicts a host CPU and one or more DSPs. It apparently is not important to the examiner that these devices are depicted separately, and that Wing So, nor Volpe, nor Bourekas teach one integrated circuit having the claimed first processor for general-purpose processing, the claimed

second processor data processing, a cache memory for the second processor, and an on-chip memory in addition to the cache memory for filling the cache memory on cache misses and for being selectably filled under user control independent of the cache misses.

Even a cursory reading of Volpe, Bourekas, and Wing So demonstrate that the claimed limitations are not taught by the argued-for combination. The rejection of claim 14 therefore fails as a matter of law. Moreover, the proffered motivation to make the argued-for combination is legally insufficient, as it states that one skilled in the art would have been motivated to make the combination because:

[P]erformance of the computer system is increased since the DSP  
microprocessor executes the CPU microprocessor operation when the CPU  
microprocessor operation is too occupied [sic]. Also, multiple waiting states are  
avoided and the blazing DSP operation speed [sic] does not come to a halt when  
interfaced to the CPU (see abstract of Wing).

(Emphasis added.)

First, claim 14 does not claim a DSP to perform CPU processing when the CPU is too occupied to do its own processing, which is the point of Wing So. Claim 14 stipulates that the first processor performs general-purpose processing and that the second processor performs data processing. Second, the avoidance of “waiting states” in Wing So has to do with offloading a portion of CPU processing to one or more external DSPs, using PCI bus communications, and has nothing to do with any particular caching arrangement.

Put simply, the examiner finds claim 14 obvious by combining a reference that does not provide any cache teachings and does not teach even the basic two-processor structure of the claimed integrated circuit, and combines it with two references that teach dramatically different single-processor caching arrangements. As such, the proffered motivation is facially invalid.

### The Dependent Claim Rejections

Claims 2-5, 7, 8, 10, 11, and 35-45 depend directly or indirectly from independent claim

1. As claim 1 is patentable over the cited references, so too are these dependent claims.

Similarly, claims 13, and 46-54 depend directly or indirectly from independent claim 12. As claim 12 is patentable over the cited references, so too are these dependent claims. Further, claims 15-18 depend directly or indirectly from independent claim 14. As claim 14 is patentable over the cited references, so too are these dependent claims.

Of course, Applicant believes that these dependent claims add further patentable limitations. Detailed rebuttals for selected ones of the dependent claims appear below, taken in approximately the same order as the dependent claim rejections are presented in the Office Action.

Claim 3 depends from claim 1 and includes the further limitation of a direct memory exchange (DME) controller operable to handle data transfers between the claimed on-chip memory and the external memory. It is error for the examiner to assert that SDC control logic 270 of Volpe teaches this limitation because, by definition, the claimed data transfers include user-directed data transfers, which the examiner has conceded that Volpe does not teach. Claim 4, which further details the DME, likewise by definition cannot be taught by Volpe.

Also regarding claim 1, claim 5 includes the limitation of a direct memory access (DMA) controller to handle storage of DMA data received via at least one DMA channel to the cache memory of the on-chip memory. Col. 3 of Volpe states that its DMA controller connects to L2 cache via a system bus interface, which cannot be argued as the claimed on-chip memory. Col. 11 of Volpe details an alternative arrangement, where the DMA controller services the prefetch buffer instead of the L2 cache. Neither arrangement teaches the claimed limitation.

Claims 7 and 8 depend from claim 5, and the arguments for claim 5 apply with equal force to them. Notably, claim 8 claims an internal memory bus connecting the DMA controller,

the on-chip memory, and a claimed cache controller, where the internal memory bus width equals the width of a line in the cache memory. To reject claim 8, the examiner combines U.S. 2004/0093479 to Ramchandran with Volpe and Bourekas. Specifically, the examiner states that paragraphs 0051 and 0052, and Fig. 7, of Ramchandran teach the limitations of claim 8. The cited sections of Ramchandran teach a cache memory, where each cache line is connected to a different memory bus through a connection having a width equal to the cache line. However, the arrangement does not describe the claimed interconnection of cache memory controller, DMA controller, and on-chip memory.

Claim 10 stipulates that the cache memory and on-chip memory of claim 1 are fabricated on the same integrated circuit die. In contrast, claim 11 depends from claim 1 and stipulates that the cache and on-chip memories are on different integrated circuit dies, but encapsulated within the same IC package. Volpe mentions nothing about die fabrication details, and provides no hint on whether one die or multiple dies are used. Despite its utter silence on any die fabrication details, the examiner finds that Volpe teaches both of these converse limitations. The examiner does so by simply pointing at functional circuits in Volpe and asserting without any evidence that Volpe teaches the limitations.

Regarding claim 35, which depends indirectly from claim 1, there is no evidentiary support for the examiner to state that Volpe teaches the claimed DME controller which selectably fills the on-chip memory of claim 1 from external memory under user control, when the entire basis for combining Bourekas with Volpe is driven by the examiner's explicit concession that Volpe does not teach filling on-chip memory under user control.

Regarding claims 38, 39, and 40, it is clear that the examiner misunderstands Volpe and Bourekas. These claims relate to the selectable control and scheduling by a user (programmer) of a DME controller that selectably fills the on-chip memory of claim 1 from external memory. Providing the claimed on-chip memory that is in some sense parallel to the claimed cache

memory allows the claimed invention to place user-controlled data within the claimed integrated circuit for immediate program access, yet does so in a manner that avoids disrupting or displacing ongoing cache operations. Bourekas explicitly displaces cache lines by writing user-data into the primary cache, and therefore has to have special “hint” data that tells its cache controller to ignore or not ignore cache hits. Ignoring cache hits is not contemplated by Volpe, nor does Volpe contemplate allowing a user to anticipate needed data and store that data in Volpe’s prefetch buffer. The rejection of claims 38, 39, and 40 completely glosses over the irreconcilable operational differences between Volpe and Bourekas.

Regarding claims 43-47 (under claim 1) and claims 52-54 (under claim 12), the examiner has conceded that Volpe does not teach transfers from external data into on-chip memory under user control. By definition, Volpe cannot teach these limitations, even when the additional teachings of Bourekas are considered.

Regarding claims 41, 42, 51, and 52, the examiner rejects these claims over the base combination of Volpe and Bourekas, in further view of U.S. 2005/0025315 to Kreitzer. These rejected claims include limitations to the integrated circuit of claim 1 (claims 40 and 42) or of claim 12 (claims 51 and 52), but set in the context of a wireless communication device.

The use of Kreitzer is self-evidently contrived and the proffered motivation to combine Kreitzer with Volpe/Bourekas is without legal or technical merit. For example, Kreitzer is exclusively directed to cryptography and securing wireless transmissions through the use of “keys.” The examiner’s rejection argument correctly indicates that Kreitzer discloses cache memory, but ignores the fact that Kreitzer depicts its cache memory 14—see, e.g., Fig. 1—outside of processor/controller 12, and outside of decryption engine 24 and encryption engine 26. Moreover, Kreitzer discloses using its “cache” memory 14 to store phonebook entries (recent calls) and/or to store encryption keys. See, e.g., Fig. 4 and the corresponding text of Kreitzer. In that sense, there is strong evidence that Kreitzer does not disclose cache memory



for caching data or instructions for immediate processor access, to avoid external memory wait states, but rather to keep handy static data related to cryptography operations. In other words, the label “cache memory” in Kreitzer appears to be used in a different, incompatible sense than the use of that label in either Bourekas or Volpe. One skilled in the art would not perceive any relevance in Kreitzer with respect to Bourekas or Volpe, and therefore would not be motivated to combine them.

Regarding claim 13, which depends from independent claim 12 and teaches DMA controller limitations similar to claim 5, it is without legal and technical merit to argue that Volpe teaches this limitation. The examiner has conceded that Volpe does not teach the claimed on-chip memory.

Regarding claims 15-18, which depend from independent claim 14, the above arguments make clear that the combination of Volpe-Bourekas-Wing So does not teach the integrated circuit of claim 14, having the first processor for general purpose processing and the second processor for data processing, and the cache and on-chip memory for the second processor.

Against the context of claim 14, claim 15 stipulates a second cache memory that is operative to store *instructions* for the processor core of the second processor, and where this second cache memory is automatically filled with *instructions* from the on-chip memory on cache misses. Thus, the integrated circuit of claim 15 includes a first cache memory that is filled with *data* from the claimed on-chip memory responsive to data misses on the first cache by the second processor, and a second cache memory that is filled with instructions from the on-chip memory responsive to instruction misses on the second cache by the second processor. Further, against this operational context, claim 15 additionally stipulates that the on-chip memory also is selectably filled with data from external memory under user control, independent of cache misses.

The examiner's rejection arguments and motivation-to-combine statements made against claim 15 mix the radically different architectures of Wing So, Bourekas, and Volpe, ignore the fact that Wing So does not teach a two-processor integrated circuit, and completely gloss over the data-versus-instruction distinctions made for the first and second cache memories in claim 15. Claims 16-18 depend from claim 15 and add further limitations not taught by the references. These rejections self-evidently fail as a matter of law.

Closing

The base combination of Volpe and Bourekas fails to teach the independent claim limitations alleged by the examiner. This failing infects all claim rejections and, consequently, all claim rejections fail as a matter of law. Further, the examiner's statements regarding the motivation to combine Volpe with Bourekas (and Volpe/Bourekas with any other reference identified in the Office Action) are conclusory and not supported by the evidentiary record. In short, the proffered motivations to make the argued-for obviousness combinations fail as a matter law.

Applicant believes that all pending claims stand in condition for immediate allowance. Reconsideration as such is respectfully requested.

Respectfully submitted,

Dated: August 7, 2007

/John Luigi Ciccozzi/  
John L. Ciccozzi  
Registration No.: 48,984  
  
5775 Morehouse Drive  
San Diego, CA 92121  
Telephone: (858) 845-2611